

Table of Contents

Bir Bilgisayarda Birden Fazla Github Hesabı Kullanımı	3
<i>Tüm hesaplar için SSH Anahtarlarının Oluşturulması</i>	3
<i>SSH Anahtarlarının SSH Agent'a Eklenmesi</i>	4
<i>SSH Public Anahtarının Github'a Eklenmesi</i>	4
<i>Config Dosyası oluşturma ve hazırlama</i>	4
<i>Github Repolarını Farklı Hesaplar Kullanarak Klonlama</i>	4

Bir Bilgisayarda Birden Fazla Github Hesabı Kullanımı

Windows işletim sistemi için yazıyorum ancak mantık diğer işletim sistemlerinde de çalışmalı. 5 basit aşama ile istediğimiz zaman hesaplar arası geçiş yapabileceğiz.

1. Tüm hesaplar için SSH anahtarları oluşturulması
2. SSH anahtarlarının SSH Agent'a eklenmesi
3. SSH public anahtarların Github'a eklenmesi
4. Ayar dosyası oluşturulması ve Host girdilerinin hazırlanması
5. Github repoları kopyalanırken farklı hesapların kullanılması

Tüm hesaplar için SSH Anahtarlarının Oluşturulması

Windows için, kullanıcı klasöründe olmanız gerekmektedir. Ancak Git Bash komut satırını kullanacağımızdan bunu dert etmemize gerek yok. Unix kodları kullanarak işlerimizi halledebiliriz.

```
C:\Users\__SENIN_KULLANICI_ADIN__\
```

1. Git Bash'i aç ve sırasıyla aşağıdaki kodları yaz.

```
cd ~
mkdir .ssh
# Klasör zaten varsa yukarıdaki hata verecektir. Varsa zaten o klasörü kullanıcaz.
cd .ssh
eval `ssh-agent -s`
```

1. Aşağıdaki kodu kullanarak her hesabın için ilgili klasörde ssh anahtarları oluştur.

```
ssh-keygen -t rsa -C "__MAIL_ADRESIN__" -f "__GITHUB_KULLANICI_ADIN__"
```

Buradaki -C anahtarı tanımlamaya yardımcı amaçlı yorum içindir. -f ise anahtarın kaydedileceği dosyanın ismidir.

Dikkat: Burada yazacağın mail adresinde github'ın bize tanımladığı mail atlamaz mail adresi olmalıdır. Zorunlu değil ancak asıl mail adresimizi karışıklık olmaması için hiçbir git işleminde kullanmamak daha iyidir.

Anahtarları ekledikten sonra **.ssh** klasörü içerisinde her hesap için bir private bir public dosya oluşması lazım.

Public anahtar **.pub** uzantısına sahip olacak ve private anahtar ise bir uzantıya sahip olmayacaktır. İkisinde ismi -f'den sonra yazdığınla aynı olacaktır.

Tüm hesapların için birer anahtar oluşturduktan sonra devam edebilirsiniz.

SSH Anahtarlarının SSH Agent'a Eklenmesi

Komut satırında hâla .ssh içerisindeyken aşağıdaki komutla anahtarları agent'a ekleyebilirsiniz.

```
ssh-add __GITHUP_KULLANICI_ADI__  
ssh-add __GITHUP_KULLANICI_ADI__
```

SSH Public Anahtarının Github'a Eklenmesi

Oluşan .pub dosyalarını herhangi bir metin editörü ile aç ve içeriğini kopyala.

<https://github.com/settings/keys> adresine giderek. **New SSH key** kısmından ekle. *Title* kısmına istediğini yazabilirsin.

Config Dosyası oluşturma ve hazırlama

1. Git Bash üzerinden `touch config` komutunu gir.
2. Yukarıdaki kod o klasörde **config** isiminde uzantsız bir dosya oluşturacaktır. O dosyayı herhangi bir metin editörü ile aç ve içerisine aşağıdakileri yapıştır.

```
# __GITHUP_KULLANICI_ADI__ account  
Host github.com-__GITHUP_KULLANICI_ADI__  
  HostName github.com  
  User git  
  IdentityFile ~/.ssh/__GITHUP_KULLANICI_ADI__  
  
# __GITHUP_KULLANICI_ADI_2__ account  
Host github.com-__GITHUP_KULLANICI_ADI_2__  
  HostName github.com  
  User git  
  IdentityFile ~/.ssh/__GITHUP_KULLANICI_ADI_2__
```

Kullanıcı adı kısımlarını kendine göre düzenle. Sayıyı istediğin kadar arttırabilirsin.

Github Repolarını Farklı Hesaplar Kullanarak Klonlama

Aşağıdaki komut ile klonlama yapabilirsin yada direkt HTTPS üzerinden klonlama yapabilirsin.

```
git clone git@github.com-  
{__GITHUP_KULLANICI_ADI__}:{__REPO_SAHIBININ_KULLANICI_ADI__}/{__REPO_ADI__}
```

.git

Commitlerin hangi hesapla yapılacağını belirlemek için her repo için ayrı ayrı kullanıcı ayarları yapman gerekmekte. Global ayarlar sadece tek bir hesap için kullanılabilir. Tüm var olan repolarına ve gelecekte klonlayacağın repolarında her zaman aşağıdaki işlemleri yap.

```
git config user.email "__GITHUP_MAIL_ADRESI__"  
git config user.name "__GITHUP_KULLANICI_ADII__"  
  
git config user.email "__GITHUP_MAIL_ADRESI_2__"  
git config user.name "__GITHUP_KULLANICI_ADII_2__"
```

Elbette sayıyı arttırabilirsin.

Son olarak **pull** ve **push** kullanabilmek için remote origin eklememiz gerekmekte. Aşağıdaki komutu her repon için çalıştır.

```
git remote add origin git@github.com-  
__GITHUP_KULLANICI_ADII__:__GITHUP_KULLANICI_ADII__  
  
git remote add origin git@github.com-  
__GITHUP_KULLANICI_ADII_2__:__GITHUP_KULLANICI_ADII_2__
```

Kaynak:

<https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

UCH Wiki'den alınmıştır.

From:
<https://wiki.ulascemh.com/> - UCH

Permanent link:
<https://wiki.ulascemh.com/doku.php?id=tr:cs:devtools:git:multipleaccounts>

Last update: **2026/04/05 12:38**

